



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/723,979	11/26/2003	David B. Gilgen	RSW9-2003-0248US1 (7161-1)	9498
46320	7590	01/09/2008	EXAMINER	
CAREY, RODRIGUEZ, GREENBERG & PAUL, LLP			DENG, ANNA CHEN	
STEVEN M. GREENBERG			ART UNIT	PAPER NUMBER
950 PENINSULA CORPORATE CIRCLE			2191	
SUITE 3020			MAIL DATE	
BOCA RATON, FL 33487			01/09/2008	
			DELIVERY MODE	
			PAPER	

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.



UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

MAILED

JAN 09 2008

Technology Center 2100

**BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES**

Application Number: 10/723,979

Filing Date: November 26, 2003

Appellant(s): GILGEN ET AL.

Scott D. Paul Reg. No. 42,984

For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed 10/24/2007 appealing from the Office action
mailed 7/24/2007.

(1) Real Party in Interest

A statement identifying by name the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The statement of the status of claims contained in the brief is correct.

(4) Status of Amendments After Final

The appellant's statement of the status of amendments after final rejection contained in the brief is correct.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

Dahlstedt et al., Publish Number: US 2004/0133895 A1, Publish Date: July 8, 2004

Tarditil, Patent Number: US 6,625,808 B1, Date of Patent: September 23, 2003

Fu, Publish Number: US 2004/0172579 A1, Publish Date: September 2, 2004

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

Claim Rejections - 35 USC § 101

1. 35 U.S.C. 101 reads as follows:

Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

2. Claims 7-9 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

Claims 7-9 set forth a memory leak detection and reporting system that is computer program claimed as computer listings per se, i.e., the descriptions or expressions of the programs, are not physical "things." They are neither computer components nor statutory processes, as they are not "acts" being performed. Such claimed computer programs do not define any structural and functional interrelationships between the computer program and other claimed elements of a computer which permit the computer program's functionality to be realized. In contrast, a claimed computer-readable medium encoded with a computer program is a computer element which defines structural and functional interrelationships between the computer program and the rest of the computer which permit the computer program's functionality to be realized, and is thus statutory. See Lowry, 32 F. 3d at 1583-84, 32 USPQ2d at 1035 (see MPEP 2106.01 "Computer-Related Nonstatutory Subject Matter").

Claim Rejections - 35 USC § 103

3. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

4. Claims 1-3, 5-17, and 19-20 are rejected under 35 U.S.C. 103(a) as being unpatentable over Dahlstedt et al. US 2004/0133895 A1 (hereinafter Dahlstedt), in view of Tarditi US 6,625,808 B1 (hereinafter Tarditi).

Per Claim 1:

Dahlstedt teaches **A memory leak detection and reporting method** (Dahlstedt, [0007], "The invention provides a system and method for detecting memory leaks"; [0008], "a reporting mechanism that reports information about said links, for use in determining potential memory leaks") **comprising the steps of: time stamping allocated ones of resources in a resource pool** (Dahlstedt, [0009], "a memory space for storing objects, ... stamping each object with a time stamp when created"); **detecting memory leaks by inspecting individual timestamps for said allocated resources to determine whether said allocated resources have become overly idle** (Dahlstedt, [0007] "the invention provides a mechanism by which objects in the memory space of a virtual machine are analyzed as to whether they are "warm" or "cold", i.e., how long they have been persisted in memory without being accessed or referenced by another object"); **and, for each allocated resource determined to have become overly idle** (Dahlstedt, [0008], "the invention comprises a system for determining potential memory leaks in a run-time environment, ... an object temperature analyzer that determines the status of warm objects and cold objects in said memory, and the links between said warm and cold objects, and, a report mechanism that reports information about said links, for use in determining potential memory leaks", also see, FIG. 3, [0021], Dahlstedt teaches cold objects as determined to have become overly idle, "The time stamp for each object is checked against the current system time, and those objects that have a time stamp older than a particular period of time are marked as cold objects"). Dahlstedt does not explicitly teach **reporting an identifying calling code segments receiving said allocated resources; identity of a corresponding one of said calling code segments**. However, Tarditi teaches **identifying calling code segments receiving said allocated resources, reporting an identity of a corresponding one of said calling code segments**. (Tarditi, FIG. 4, call stack 402 or 406, col. 11, lines 10-49, "a live object is an object which has an identifiable pointer in the root set, e.g., call stack 402 or 406... for each transition from a GC frame to a non-GC frame in the call stack 402, creation function 304 allocates space on the stack frame for a transition record, e.g., transition records 420 and 424. The transition store select pointer and state information, as well as a pointer to the immediate past transition record, e.g., transition record

424."), the transition record read on the limitation of reporting identity of a corresponding one of said calling code segments.

It would have been obvious to one having ordinary skill in the computer art at the time of the invention was made to modify the method disclosed by Dahlstedt to include identifying calling code segments receiving said allocated resources; reporting an identity of a corresponding one of said calling code segments using the teaching of Tarditi. The modification would be obvious because one of ordinary skill in the art would be motivated to facilitate automated memory management among heterogeneous components of a computer program (Tarditi, col. 4, lines 40-42).

Per Claim 2:

The rejection of claim 1 is incorporated, Dahlstedt does not explicitly teach **the step of retrieving identities for individual ones of said calling code segments from an associated calling stack when said individual ones of said calling code segments acquire one of said allocated resources.** However, Tarditi teaches **the step of retrieving identities for individual ones of said calling code segments from an associated calling stack when said individual ones of said calling code segments acquire one of said allocated resources** (Tarditi, FIG. 6, step 606, col. 12, lines 58-63).

Per Claim 3:

The rejection of claim 1 is incorporated, Dahlstedt teaches detecting step and Tarditi teaches reporting step (see discussion above in claim 1). Dahlstedt further teaches **the step of performing said detecting and reporting steps in a separate thread of execution** (Dahlstedt, FIG. 1, steps 6-10, [0017], "in step 6, after a particular time, all of the objects are analyzed for potential memory leaks ... In step 8, all of the warm objects which are pointing to each other are clustered together in a warm cluster, and similarly all of the code objects that are pointing together are clustered together to form a cold cluster ... In step 10, an object map is displayed graphically showing the programmer software developer the links between the cold clusters and the warm clusters, and also informing the developer which particular

field in which particular warm object is referencing the cold cluster. This gives the developer a powerful graphical view of their object memory usage ..."), Dahlstedt teaches the detecting step (steps 6-8) and the reporting steps (step 10) that read on the limitation of in a separate thread of execution in the claim.

Per Claim 5:

The rejection of claim 1 is incorporated, Dahlstedt teaches detecting step and Tarditi teaches reporting step (see discussion above in claim 1). Both Dahlstedt and Tarditi further teach **the step of performing said detecting and reporting steps responsive to allocating one of said resources in said resource pool** (Dahlstedt, [0009], "said run-time environment including a virtual machine and a memory space for storing objects, comprising: creating objects in the memory of a run-time environment; stamping each object with a time stamp when created; updating the time stamp as each object is accessed or referenced; after a time T_{check} marking each object as being either warm or cold; clustering warm objects together to form warm clusters and cold objects together to form cold cluster; and displaying an object map showing links between warm clusters and cold clusters"); and (Tarditi, col. 9, lines 23-45, "when execution of the program proceeds to a creation function, the creation function creates a transition record on the stack frame of the particular thread in which the function call is identified. The transition record is populated with select pointer and state information").

Per Claim 6:

The rejection of claim 1 is incorporated, and Dahlstedt further teaches **the step of performing said detecting and reporting steps responsive to an elapsing interval** (FIG. 4, [0022], "The system is then allowed to run for check time T_{check} , in step 98. This process is repeated in step 100 for each object in memory. In step 102, the system receives a request, i.e., either an automatic request or a determined request by the operator, to view the object display or object map. In step 104, the time stamp is checked for each object in the memory... If $T_{check} - T_{access}$ is greater than a specified time T_{limit} , the object may be considered cold... all object in memory will now be indicated as being warm or cold. In step 110, all cold

objects that point to another are clustered together... These are of particular importance to the software developer since thy represent large potential memory leaks").

Per Claim 7:

Dahlstedt teaches **A memory leak detection and reporting system ([0007]) comprising: a resource pool comprising a plurality of allocable resources ([0008], "a memory space for storing objects"); a pool manager programmed to manage allocation of said allocable resources to calling code segments ([0009], "creating objects in memory of a run-time environment; stamping each object with a time stamp when created; updating the time stamp as each object is accessed or referenced"); and, a data store of allocated resources (Dahlstedt, FIG. 4, [0022]), wherein the pool manager detects memory leaks by inspecting individual timestamps for said allocated resources to determine whether said allocated resources have become overly idle (Dahlstedt, [0007]** "the invention provides a mechanism by with objects in the memory space of a virtual machine are analyzed as to whether they are "warm" or "cold", i.e., how long they have been persisted in memory without being accessed or referenced by another object"); **and, for each allocated resource determined to have become overly idle (Dahlstedt, [0008], "the invention comprises a system for determining potential memory leaks in a run-time environment, ... an object temperature analyzer that determines the status of warm objects and cold objects in said memory, and the links between said warm and cold objects, and, a report mechanism that reports information about said links, for use in determining potential memory leaks", also see, FIG. 3, [0021], Dahlstedt teaches cold objects as determined to have become overly idle, "The time stamp for each object is checked against the current system time, and those objects that have a time stamp older than a particular period of time are marked as cold objects"); Dahlstedt does not explicitly teach corresponding identities for calling code segments receiving said allocated resources; reports an identity of a corresponding one of said calling code segments to the data store. However, Tarditi teaches corresponding identities for calling code segments receiving said**

allocated resources; reports an identity of a corresponding one of said calling code segments to the data store (Tarditi, col. 10, lines 18-28).

It would have been obvious to one having ordinary skill in the computer art at the time of the invention was made to modify the method disclosed by Dahlstedt to include corresponding identities for calling code segments receiving said allocated resources; reports an identity of a corresponding one of said calling code segments to the data store using the teaching of Tarditi. The modification would be obvious because one of ordinary skill in the art would be motivated to facilitate automated memory management among heterogeneous components of a computer program (Tarditi, col. 4, lines 40-42).

Per Claim 8:

The rejection of claim 7 is incorporated, Dahlstedt does not explicitly teach **a communicative coupling to a call stack; and, correlation logic for correlating a calling code segment reference disposed in said call stack to a concurrently allocated one of said allocable resources**. However, Tarditi teaches **a communicative coupling to a call stack** (Tarditi, FIG. 3, col. 9, lines 62-67, col. 10, lines 1-10); **and, correlation logic for correlating a calling code segment reference disposed in said call stack to a concurrently allocated one of said allocable resources** (Tarditi, FIG. 3, col. 10, lines 11-28).

Per Claim 9 (Currently Amended):

The rejection of claim 7 is incorporated, and Dahlstedt further teaches **a garbage collector coupled to said resource pool** (Dahlstedt, [0007], "The invention provides a system and method for detecting memory leaks in a garbage collected environment").

Per Claim 10:

Dahlstedt teaches **A memory leak detection and reporting method** (Dahlstedt, [0007]) comprising the steps of: **allocating a resource from a resource pool, time stamping said allocated**

resource and recording an identity for a calling code segment acquiring said allocated resource (Dahlstedt, FIG. 4, [0022], "the object header includes a time stamp field which is initially set as T_{init} "); updating said time stamp when said allocated resource is accessed (Dahlstedt, [0022], "...as the object is used, or referenced by a subsequent object, the time stamp field is updated with the last access time T_{access} for this particular object"); inspecting said time stamp to determine if said allocated resource has become overly idle (Dahlstedt, FIG. 4, [0022], "The system is then allowed to run for a check time T_{check} , in step 98"); and, if it is determined that said allocated resource has become overly idle, reporting a **suspected memory leak in association with said allocated resource** (Dahlstedt, FIG. 4, [0022], "This process is repeated in step 100 for each object in memory. In step 102, the system receives a request, i.e., either an automatic request or a determined request by the operator, to view the object display or object map. In step 104, the time stamp is checked for each object in the memory... If $T_{check} - T_{access}$ is greater than a specified time T_{limit} , the object may be considered cold... all object in memory will now be indicated as being warm or cold. In step 110, all cold objects that point to another are clustered together... These are of particular importance to the software developer since they represent large potential memory leaks"); Dahlstedt does not explicitly teach **further reporting said recorded identity for said calling code segment which had acquired said allocated resource**. However, Tarditi teaches **further reporting said recorded identity for said calling code segment which had acquired said allocated resource** (Tarditi, col. 13, lines 16-25).

It would have been obvious to one having ordinary skill in the computer art at the time of the invention was made to modify the method disclosed by Dahlstedt to include further reporting said recorded identity for said calling code segment which had acquired said allocated resource using the teaching of Tarditi. The modification would be obvious because one of ordinary skill in the art would be motivated to facilitate automated memory management among heterogeneous components of a computer program (Tarditi, col. 4, lines 40-42).

Per Claim 11:

The rejection of claim 10 is incorporated, Dahlstedt does not explicitly teach **the step of performing said inspecting and reporting steps in a separate thread of execution**. However, Tarditi teaches **the step of performing said inspecting and reporting steps in a separate thread of execution** (Tarditi, col. 11, lines 3-5).

Per Claim 12:

The rejection of claim 10 is incorporated, and Dahlstedt further teaches **the steps of performing said allocating, inspecting and reporting steps in a pool manager** (Dahlstedt, [007]-[0008]).

Per Claim 13:

The rejection of claim 10 is incorporated, and Dahlstedt further teaches **the step of performing said inspecting and reporting steps responsive to allocating another resource in said resource pool** (Dahlstedt, [0007]-[0008]).

Per Claim 14:

The rejection of claim 10 is incorporated, and Dahlstedt further teaches **the step of performing said inspecting and reporting steps responsive to an elapsing interval** (Dahlstedt, FIG. 3, [0021], lines 13-29).

Per Claims 15-17, and 19:

These claims are the machine readable storage version of the claimed method discussed above (claims 1-3, and 6), wherein all claim limitations also have been addressed and/or covered in cited areas as set forth above. Thus, accordingly these claims are also obvious.

Per Claim 20:

This claim is the machine readable storage version of the claimed method discussed above (claim 10), wherein all claim limitations also have been addressed and/or covered in cited areas as set forth above. Thus, accordingly this claim is also obvious.

5. Claims 4 and 18 are rejected under 35 U.S.C. 103(a) as being unpatentable over Dahlstedt et al. US 2004/0133895 A1 (hereinafter Dahlstedt), in view of Tarditi US 6,625,808 B1 (hereinafter Tarditi), and further in view of Fu US 2004/0172579 A1 (hereinafter Fu).

Per Claim 4:

The rejection of claim 2 is incorporated, the combination of Dahlstedt and Tarditi does not explicitly teach **the steps of: for calling code segment in which a resource is allocated, inducing a placebo error condition in close proximity to code for allocating said resource; and, reading error data from said calling stack produced by said placebo error condition, said error data comprising identity information for said calling code segment**. However, Fu teaches teach **the steps of: for calling code segment in which a resource is allocated, inducing a placebo error condition in close proximity to code for allocating said resource** (Fu, FIGS. 5-6, and related text [0042]-[0047], "in decision block 530, a determination is made whether minima point processing subroutine 600 returned an indication of a memory leak. If a memory leak was found, processing proceeds to block 599 where the memory usage processing subroutting 500 ends and a memory leak message is returned to the calling routing ... subroutine 600 proceeds to decision block 610 where a test is made to determine whether at least four memory usage data minima point were found. If less that four memory leak message data minima points were found, processing proceeds to block 699, where subroutine 600 ends and a memory leak message is returned to the calling routine"); **and, reading error data from said calling stack produced by said placebo error condition, said error data comprising identity information for said calling code segment** (Fu, [0044]).

It would have been obvious to one having ordinary skill in the computer art at the time of the invention was made to modify the method disclosed by the combination of Dahlstedt and Tarditi to include the steps of: for calling code segment in which a resource is allocated, inducing a placebo error condition in close proximity to code for allocating said resource; and, reading error data from said calling stack produced by said placebo error condition, said error data comprising identity information for said calling code segment using the teaching of Fu. The modification would be obvious because one of ordinary skill in the art would be motivated to detecting memory leaks in computing environments that does not require examining all aspects of a computer's memory (Fu, [0003], lines 22-24).

Per Claim 18:

This claim is the machine readable storage version of the claimed method discussed above (claim 4), wherein all claim limitations also have been addressed and/or covered in cited areas as set forth above. Thus, accordingly this claim is also obvious.

(10) Response to Arguments

(1) The Rejection of Claims 7-9 Under 35 U.S.C. §101

Applicant argued:

Software alone is incapable of doing anything because it is disconnected from hardware. Software alone is also covered under M.P.E.P. §2106.01 with regard to "functional descriptive material." Claims 7-9, however, are not directed to "functional descriptive material." Instead, claims 7-9 are directed to a system...

Despite the Examiner's continued failure to set forth an explanation as to why these claimed elements are directed to an abstract idea, reference is made to the recent decision of the Federal Circuit of In re Comiskey. ...

Examiner response:

Applicants specify the system in claims 7-9 does not have required hardware and expected state, thus, it can be a software system. Therefore, it is "functional descriptive material", and rejected under 35 U.S.C. 101.

In response to In re Comiskey, it is still under re-hearing.

(2) The Rejection of Claims 1-3, 5-17, and 19-20 Under 35 U.S.C. §103 for Obviousness based upon Dahlstedt in view of Tarditi

Appellants argued:

Claim 1

The Examiner's analysis, however, fails to consider the claimed invention, as a whole, specifically, claim 1 recites:

for each allocated resource determined to have become overly idle, reporting an identity of a corresponding one of said calling code segments (emphasis added).

As readily evident upon reviewing this passage, the second clause (i.e., the underlined clause) is conditioned upon the first clause (i.e., the *italicized* clause). Thus, there is an inextricable relationship between the first clause and the second clause.

Examiner response:

In response to applicant's arguments against the references individually, one cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references. See *In re Keller*, 642 F.2d 413, 208 USPQ 871 (CCPA 1981); *In re Merck & Co.*, 800 F.2d 1091, 231 USPQ 375 (Fed. Cir. 1986). Moreover, Dahlstedt teaches a report mechanism that reports information about said links (between said warm and cold object in said memory) for use in determining potential memory leaks (Dahlstedt, [0008]). Dahlstedt dose not explicitly teach reporting the

an identity of a corresponding one of said calling code segment. However, Tarditi teaches reporting an identity of a corresponding one of said calling code segments (Tarditi, FIG. 4, call stack 402 or 406, col. 11, lines 10-49, "a live object is an object which has an identifiable pointer in the root set, e.g., call stack 402 or 406... for each transition from a GC frame to a non-GC frame in the call stack 402, creation function 304 allocates space on the stack frame for a transition record, e.g., transition records 420 and 424. The transition store select pointer and state information, as well as a pointer to the immediate past transition record, e.g., transition record 424", emphasis added), Tarditi teaches the transition records read on the limitation of reporting an identity of a corresponding one of said calling code segments. Thus, the combination of Dahlstedt and Tarditi does present logical consistency and teaches all the limitations in claim 1 of the present application.

Appellants argued:

Claim 3

Appellants are unclear as to where, specifically, Tarditi teaches that the detecting and reporting steps are performed in separate thread of execution.

Examiner response:

The rejection of claim 1 is incorporated, Dahlstedt teaches detecting step (Dahlstedt, [0007] "the invention provides a mechanism by with objects in the memory space of a virtual machine are analyzed as to whether they are "warm" or "cold", i.e., how long they have been persisted in memory without being accessed or referenced by another object") and Tarditi teaches reporting step (Tarditi, FIG. 4, call stack 402 or 406, col. 11, lines 10-49, "a live object is an object which has an identifiable pointer in the root set, e.g., call stack 402 or 406... for each transition from a GC frame to a non-GC frame in the call stack 402, creation function 304 allocates space on the stack frame for a transition record, e.g., transition records 420 and 424. The transition store select pointer and state information, as well as a pointer to the immediate past transition record, e.g., transition record 424."), Dahlstedt further teaches **the step of**

performing said detecting and reporting steps in a separate thread of execution (Dahlstedt, FIG. 1, steps 6-10, [0017], "in step 6, after a particular time, all of the objects are analyzed for potential memory leaks ... In step 8, all of the warm objects which are pointing to each other are clustered together in a warm cluster, and similarly all of the code objects that are pointing together are clustered together to form a cold cluster ... In step 10, an object map is displayed graphically showing the programmer software developer the links between the cold clusters and the warm clusters, and also informing the developer which particular field in which particular warm object is referencing the cold cluster. This gives the developer a powerful graphical view of their object memory usage ..."), Dahlstedt teaches the detecting step (steps 6-8) and the reporting steps (step 10) read on the limitation of "in a separate thread of execution" in claim 3.

Appellants argued:

Claim 5

As already noted above, however, Dahlstedt fails to teach the claimed reporting step. Thus, Dahlstedt cannot teach all the limitations of claim 5.

Examiner response:

Again, in response to applicant's arguments against the references individually, one cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references. See *In re Keller*, 642 F.2d 413, 208 USPQ 871 (CCPA 1981); *In re Merck & Co.*, 800 F.2d 1091, 231 USPQ 375 (Fed. Cir. 1986).

The rejection of the base claim (claim 1) is incorporated; Dahlstedt teaches detecting step and Tarditi teaches reporting step (see discussion above in claim 1). Both Dahlstedt and Tarditi further teach **the step of performing said detecting and reporting steps responsive to allocating one of said resources in said resource pool** (Dahlstedt, [0009], "said run-time environment including a virtual machine and a memory space for storing objects, comprising: creating objects in the memory of a run-time environment; stamping each object with a time stamp when created; updating the time stamp as

each object is accessed or referenced; after a time T_{check} marking each object as being either warm or cold; clustering warm objects together to form warm clusters and cold objects together to form cold cluster; and displaying an object map showing links between warm clusters and cold clusters"); and (Tarditi, col. 9, lines 23-45, "when execution of the program proceeds to a creation function, the creation function creates a transition record on the stack frame of the particular thread in which the function call is identified. The transition record is populated with select pointer and state information").

Dahlstedt's teaching "creating objects in the memory of a run-time environment" read on the limitation of **allocating one of said resource resources in said resource pool**, "updating the time stamp as each object is accessed or reference, ... and displaying an object map ..." read on the limitation of **detecting and reporting steps responsive to allocating one of said resources in said resources in said resource pool**; and Tarditi's teaching "the creation function creates a transition record on the stack frame of the particular thread in which the function call is identified. The transition record is populated with select pointer and state information" that also read on the limitations of **the step of performing said detecting and reporting steps responsive to allocating one of said resources in said resource pool** in claim 5 of the present application.

(3) The rejection of claims 4 and 18 under 35 U.S.C. §103 for obviousness based upon Dahlstedt in view of Tarditi and Fu

Appellants argued:

Applicants are unable to determine what specific teaching in Fu is being relied upon to teach claimed "inducing a placebo error condition in close proximity to code for allocating said resource".

Examiner response:

Fu teaches teach **inducing a placebo error condition in close proximity to code for allocating said resource** (Fu, FIGS. 5-6, and related text [0042]-[0047], "in decision block 530, a determination is made whether minima point processing subroutine 600 returned an indication of a

memory leak. If a memory leak was found, processing proceeds to block 599 where the memory usage processing subrouting 500 ends and a memory leak message is returned to the calling routine ... subroutine 600 proceeds to decision block 610 where a test is made to determine whether at least four memory usage data minima point were found. If less than four memory leak message data minima points were found, processing proceeds to block 699, where subroutine 600 ends and a memory leak message is returned to the calling routine");

In FIGS 5-6 and the related text, Fu basically teaches return an indication of a memory leak in the memory usage to the calling routine (calling code segment) that read on the limitations in claims 4 and 18 in the present application.

(11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

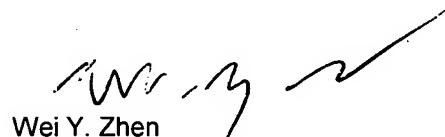
For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

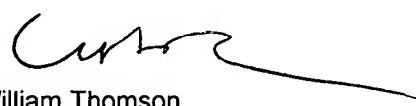


Anna Deng

Conferees:


Wei Y. Zhen

WEI ZHEN
SUPERVISORY PATENT EXAMINER


William Thomson